

# DEFORMING DEVICE AND GENERATING DEVICE FOR NETWORK

Publication number: JP4218872

Publication date: 1992-08-10

Inventor: TANAKA HIROSHI

Applicant: FUJITSU LTD

Classification:

- International: G06F17/27; G06F17/30; G10L13/04; G10L15/10; G10L15/18;  
G06F17/27; G06F17/30; G10L13/00; G10L15/00; (IPC1-7):  
G06F15/40

- European: G10L15/18C1

Application number: JP19900412079 19901219

Priority number(s): JP19900412079 19901219

Also published as:

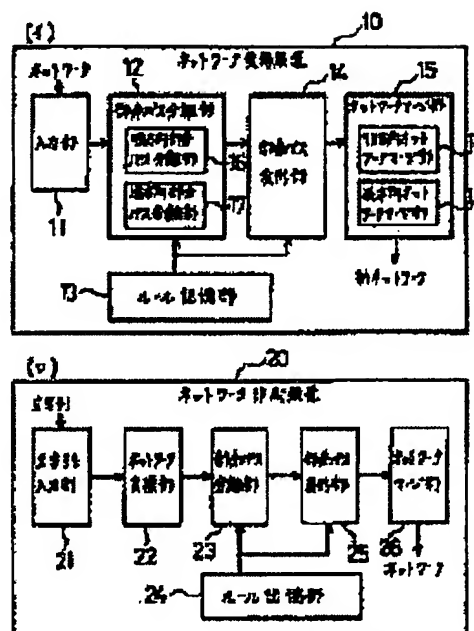


US5345537 (A1)

Report a data error here

## Abstract of JP4218872

**PURPOSE:** To generate a network with low redundancy and high efficiency for a deforming device and a generating device for network which applies a rewrite rule to the network and generates a new network after rewrite. **CONSTITUTION:** A partial pass in the network targeted to deform or the one converted from a character string is separated by a partial pass separation part 12. A separated partial pass is deformed according to the rewrite rule stored in a rule storage part 13. A common part between the partial pass after deformation and a remaining network from which the partial pass is separated is merged by a network merge part 15. A merged network is set as the new network.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平4-218872

(43) 公開日 平成4年(1992)8月10日

(51) Int.Cl.<sup>5</sup>

G 0 6 F 15/40

識別記号

庁内整理番号

F I

技術表示箇所

5 0 0 N 7056-5L

審査請求 未請求 請求項の数5 (全 10 頁)

(21) 出願番号 特願平2-412079

(22) 出願日 平成2年(1990)12月19日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 田中 宏

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

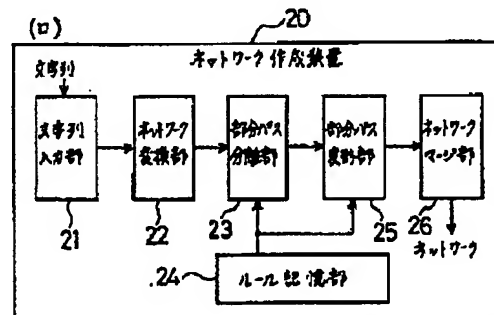
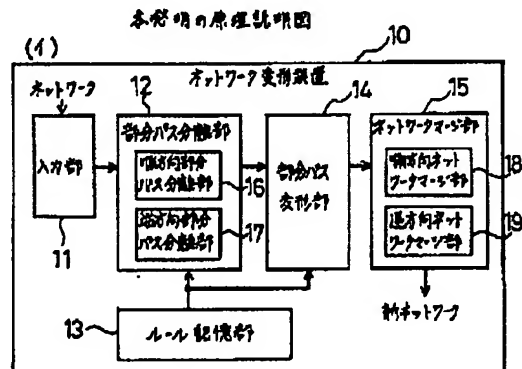
(74) 代理人 弁理士 小笠原 吉義 (外2名)

(54) 【発明の名称】 ネットワーク変形装置および作成装置

(57) 【要約】

【目的】 ネットワークに書き換えルールを適用して、書き換え後の新ネットワークを作成するネットワーク変形装置および作成装置に関し、冗長性の少ない効率のよいネットワークを生成することを目的とする。

【構成】 変形対象となるネットワークまたは文字列から変換したネットワーク中の部分バスを、部分バス分離部12によって分離する。その分離された部分バスをルール記憶部13に記憶している書き換えルールに従って変形する。変形後の部分バスと、部分バスが分離された残りのネットワークとの共通部分を、ネットワークマージ部15によってマージする。マージされたネットワークを、新ネットワークとする。



1

## 【特許請求の範囲】

【請求項1】 データの基本単位を接続したネットワークを、所定の書き換えルールに従って計算機により変形するネットワーク変形装置において、変形対象となるネットワークを入力する入力部(11)と、書き換えルールを記憶するルール記憶部(13)と、書き換えルールの書き換え元にマッチする変形対象ネットワーク中の部分パスを分離する部分パス分離部(12)と、分離された部分パスを書き換えルールに従って変形する部分パス変形部(14)と、変形後の部分パスと、その部分パスが分離された残りのネットワークとの共通部分をマージするネットワークマージ部(15)とを備えたことを特徴とするネットワーク変形装置。

【請求項2】 前記部分パス分離部(12)は、前方ノードから順に部分パスを分離していく順方向部分パス分離部(16)と、後方ノードから順に部分パスを分離していく逆方向部分パス分離部(17)とを有することを特徴とする請求項1記載のネットワーク変形装置。

【請求項3】 前記ネットワークマージ部(15)は、前方ノードから順にマージを行う順方向ネットワークマージ部(18)と、後方ノードから順にマージを行う逆方向ネットワークマージ部(19)とを有することを特徴とする請求項1または請求項2記載のネットワーク変形装置。

【請求項4】 前記ネットワークは、音声認識時に用いる辞書に登録される音響セグメントネットワークであることを特徴とする請求項1、請求項2または請求項3記載のネットワーク変形装置。

【請求項5】 データの基本単位を接続したネットワークを、入力文字列から所定の書き換えルールに従って計算機により作成するネットワーク作成装置において、ネットワークのもととなる文字列を入力する文字列入力部(21)と、入力した文字列をネットワークに変換するネットワーク変換部(22)と、書き換えルールを記憶するルール記憶部(24)と、前記ネットワーク変換部(22)が出力したネットワークを変形対象とし、書き換えルールの書き換え元にマッチする変形対象ネットワーク中の部分パスを分離する部分パス分離部(23)と、分離された部分パスを書き換えルールに従って変形する部分パス変形部(25)と、変形後の部分パスと、その部分パスが分離された残りのネットワークとの共通部分をマージするネットワークマージ部(26)とを備えたことを特徴とするネットワーク作成装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、ネットワークに書き換えルールを適用して、書き換え後の新しいネットワークを作成するネットワーク変形装置および作成装置に関する。例えば、音声認識あるいは音声合成装置等では、「音響セグメントネットワーク」または「発音ネットワーク」等と呼ばれるネットワークを利用する手法が用い

2

られている。これらのネットワークを効率よく変形または作成する技術が必要とされる。このようなネットワークの音声認識における利用法やその有効性は、例えば次の文献に詳しく述べられている。

「"100,000-Word Recognition Using Acoustic-Segment Networks", IEEB ICASSP90, Apr, 1990」

## 【0002】

【従来の技術】 音声認識や音声合成の手法として、単語等の文字列を発音列で表した音響セグメントネットワークを使用する技術が開発されつつある。図11は、音響セグメントネットワークを用いた音声認識システムの説明図である。まず、入力音声100を音響処理部101に入力する。音響処理部101では、各種の発音の特徴を格納した音響テンプレート102と、入力音声100のマッチングをとり、入力音声100を発音の候補列に変換する。次に、音響処理部101で求めた入力音声100の発音候補列を照合部103に入力する。照合部103は、単語等を発音ネットワークで表した音響セグメントネットワーク104と時間長ルール105を使用して、入力音声100の発音候補列を意味のある単語として認識し、認識結果106を出力する。

【0003】 図12は、その音響セグメントネットワークの例であり、同図(ハ)は、「アサヒ」という単語の音響セグメントネットワークを示している。#は単語の開始と終了を表し、その間の発音の変化を、音素のデータを接続したネットワークで表している。例えば「#-A-S-A-H-I-#」という経路も、「#-A-S-u-A-H-I-#」という経路も、「アサヒ」という単語の発音である。これにより、人によって異なる発音の仕方などを吸収することができる。

【0004】 音声認識でこの音響セグメントネットワークを使用するためには、認識する単語についてあらかじめ図12の(ハ)に示すようなネットワークを作成しておく必要がある。認識単語数が少ないならば、それぞれの単語についての音響セグメントネットワークを一つ一つ人手で作ることも可能である。

【0005】 しかし、一般に使用する言葉の数を認識しようとする場合には、その単語数は膨大で、それらの音響セグメントネットワークを人手で作ることは、現実的には不可能である。そこで、例えば図12の(イ)に示すような単語の文字列を入力し、それを(ロ)に示すように単純なネットワークに変換した後、所定の書き換えルールを用いることにより、(ハ)に示す音響セグメントネットワークを、自動的に作成することが考えられている。

【0006】 図13は、その従来技術の説明図である。枝分かれのないネットワークに、書き換えルールを適用するのは容易である。枝分かれのないネットワークというのは、例えば図13の(イ)に示すnet1のようなものである。なお、「-」は文字間のつながりを示すもので

あり、文字としての意味は持たない。このネットワーク net1 は、文字列 Q, B, C, A, R. と等価である。

【0007】書き換えルールには、書き換え先が一つしかないもの (rule1, rule2) および複数の書き換え先があるもの (rule3, rule4) がある。また、文脈自由であるもの (rule1, rule3) と、文脈依存であるもの (rule2, rule4) という区分もある。

【0008】ルール rule1 は、ネットワーク中に A. という文字があったならば、B. に書き換えるという意味である。net1 に適用すると、net2 ができる。ルール rule 2 は、“/”以降が文脈を表しており、“\_”が変形対象となるボタン（この場合、A.）の位置を表す。すなわち、C. と R. に挟まれた A. があったならば、B. に書き換えるという意味である。これを net1 に適用すると、同様に net2 になる。ただし、例えば net1 の最後の文字が、R. ではなく S. である場合には、文脈が異なるため、書き換えは行われない。

【0009】ルール rule3 は、A. という文字を B. と C. に書き換えるというものである。このように書き換え可能性が複数である場合には、枝分かれのあるネットワークに変形される。例えば、net1 に rule3 を適用すると、net3 ができる。同様に、rule4 は、文脈まで一致した場合にのみ、上記のような書き換えを実行する。

【0010】次に、net3 のような枝分かれのあるネットワークにルールを適用する場合について、図 13 の (ロ) に従って説明する。ルール rule5 は、C. B. というつながりを、Z. に書き換えるものである。これを (イ) に示す net3 に適用すると、本来は net4 のようになる。しかしながら、net3 の C. B. を単に Z. に書き換えただけでは、もう一つの C. につながっていた枝が、net5 に示すように、つなぎ先を失ってしまう可能性がある。この方法では、正しいネットワークを得ることはできない。

【0011】さて、上記の問題を背景として、一つの解決策が提案されている。net5 に示すような枝のつなぎ先が失われる問題は、枝分かれのあるネットワークに直接ルールを適用しようとしたことに起因する。そこで、枝分かれのあるネットワークを、枝分かれのない複数のネットワークに分割し、それぞれにルールを適用した後、再び一つのネットワークに戻すというのが提案されている解決策である。

【0012】まず、図 13 の (イ) に示す net1 に、rule 3 (または rule4) を適用すると、net3 ができる。これを、このような形でなく、(ハ) に示す net6, net7 のような 2 つの枝分かれのないネットワークで表すようにする。これらに、それぞれ (ロ) に示す rule5 を適用すると、net6 が net8 に変形される。最後に net7 と net8 を合成することにより、正しいネットワークである net4 ができる。

【0013】図 14 は、ネットワークの合成説明図である。二つのネットワークを合成する手段としては、まず

それぞれのネットワークのノードの間の対応関係を調べる。なお、ノードとは、A., Q. などのネットワークを構成する最小単位のことをいう。対応関係の調べ方としては、動的計画法 (DP 法) を用いる方法が提案されている。

【0014】net7 と net8 の対応関係は、図 14 の (イ) に示すようになる。次にこれらに基づき、対応のついたノード同士を一つにまとめて、接続関係を書き直す。その結果を、図 14 の (ロ) に示す。これは、図 13 の (ロ) に示す目的とする結果の net4 に他ならない。

【0015】

【発明が解決しようとする課題】さて、次のようなケースについて、従来技術として説明した方法の適用を考えてみる。図 15 に示す例において、net9 に rule6 ~ rule 8 を順に適用した場合、ネットワークは、本来、net10 のようにならなければならない。これを前述した方法で作る場合、途中結果として枝分かれのないネットワーク net11, net12 が二つできる。さらにこの二つのネットワーク net11, net12 を一つに合成する。

【0016】そのため、まず図 16 の (イ) に示すように、各ネットワーク net11, net12 の対応関係を調べる。次にこれらを一つのネットワークにマージすると、図 15 に示す net13 ができる。これを本来あるべき net10 と比較すると、A, U, C, Z, E. および A, X, C, Y, E. のパスが余分にできていることがわかる。

【0017】以上のように、従来考えられている方法をそのまま用いた場合、本来は存在しないようなパスが新たに作られる可能性がある。音声認識装置を実現しようとした場合に、このような例が「音響セグメントネットワーク」作成時に発生すると、装置の認識性能の悪化につながる。

【0018】上記のような余分なパスの生成を抑える方法として、一つの方法が容易に考えられる。すなわち、二つのネットワークをマージする際に、ネットワークの始端・終端から順に両ネットワークの比較を行い、まったく同じである部分だけをマージする方法である。要するに、二つのネットワークを両端から絞るようにしてマージすることになる。

【0019】net11, net12 を対応づける場合、両端からネットワークの比較をしていくと、先頭の A. と最終の E. だけがマージできるノードだということになる。したがって、真ん中の C. 同士の対応は無効となる。こうして、図 16 の (ロ) に示すような対応づけに従ってマージする。これに従ってマージすれば、net10 のようなネットワークが作られる。

【0020】しかし、この方法にも問題点がある。例えば図 15 に示す net9 に対して、図 15 に示す rule6 および図 16 の (ハ) に示す rule9 を適用する場合である。このとき、途中結果として図 16 の (ハ) に示す net14 ~ net17 ができ、これらをマージすることによって目的

5

のネットワークが作られる。

【0021】これらを正しくマージすると、図15に示すnet13ができるはずである。しかしながら、前述した方法に基づいた解決法を用いると、まずnet14とnet15とをマージすることによって、図16の(二)に示すようにnet18が作られ、次にnet18とnet16とをマージすることによって、図16の(ホ)に示すようにnet19が作られ、最後にnet19とnet17とをマージすることによって、図16の(ハ)に示すnet20が作られる。

【0022】以上のケースでは、net13とnet20とを比べると明らかなように、正しいネットワーク(net13)と等価であるとはいえず、非常に大きなネットワーク(net20)が作られることになる。すなわち、ネットワークが冗長になったといえる。ネットワークの冗長性は、これを音声認識装置に用いた場合の記憶容量の増大につながるだけでなく、処理速度の低下にもつながる。

【0023】以上のように従来技術を用いると、適当なネットワークを生成することができないことがあり、例えば音声認識装置を実現しようとする場合に、装置の性能を劣化させる原因となるという問題がある。本発明は上記問題点の解決を図り、冗長性の少ないネットワークを効率よく作成する手段を提供し、そのネットワークを用いる装置の記憶容量の削減および処理速度の向上を可能とすることを目的とする。

【0024】

【課題を解決するための手段】図1は、本発明の原理説明図である。図1の(イ)に示すネットワーク変形装置10は、CPUおよびメモリを備え、データの基本単位を接続したネットワークを、所定の書き換えルールに従って目的とするネットワークに変形する装置である。入力部11は、その変形対象となるネットワークを入力する。ルール記憶部13には、ネットワークを変形するための書き換えルールが記憶されている。

【0025】部分バス分離部12は、ルール記憶部13が記憶している書き換えルールに基づいて、入力部11が入力した変形対象ネットワーク中の部分バスを分離する。すなわち、書き換えルールの書き換え元にマッチする部分バスを、変形対象ネットワーク中から分離する。部分バス変形部14は、部分バス分離部12によって分離された部分バスを、書き換えルールに従って変形する。

【0026】ネットワークマージ部15は、部分バス変形部14による変形後の部分バスと、その部分バスが分離された残りのネットワークとの共通部分をマージし、マージ結果を新しいネットワークとして出力する。

【0027】部分バス分離部12は、前方ノードから順に部分バスを分離していく順方向部分バス分離部16と、後方ノードから順に部分バスを分離していく逆方向部分バス分離部17とを有し、書き換えルールの書き換え元にマッチする部分バスを、前方ノードおよび後方ノ

6

ードからのマッチングによって分離する。

【0028】ネットワークマージ部15は、前方ノードから順にマージを行う順方向ネットワークマージ部18と、後方ノードから順にマージを行う逆方向ネットワークマージ部19とを有し、それぞれ部分バスと残りのバスにおける対応するノードを合成する。

【0029】書き換えルールによって変形するネットワークは、あるデータの基本単位を、ポイントまたは識別番号などの何らかのリンク手段によって接続した構造を持つものであり、このような構造を持つデータであれば、本発明を適用することができる。特に、音声認識時に用いる辞書に登録される音響セグメントネットワークの作成・変形に適用すれば、記憶容量および処理速度の面で、効率のよいネットワークを得ることができる。

【0030】図1の(ロ)に示すネットワーク作成装置20は、CPUおよびメモリなどを備え、データの基本単位となる文字(単なるコード情報を含む)の列を入力し、その入力文字列から所定の書き換えルールに従ってネットワークを作成する装置である。

【0031】ネットワーク作成装置20は、ネットワークのもととなる文字列を入力する手段として文字列入力部21を持つ。ネットワーク変換部22は、文字列入力部21が入力した文字列を、各文字の接続関係を示すネットワーク構造を持つデータに変換する。すなわち、入力文字列中の各文字を順番につなぐことにより、ネットワークを生成する。

【0032】部分バス分離部23、ルール記憶部24、部分バス変形部25およびネットワークマージ部26は、それぞれ図1の(イ)に示す部分バス分離部12、ルール記憶部13、部分バス変形部14、ネットワークマージ部15と同じ機能を持つ。図1(イ)に示すネットワーク変形装置10が、ネットワークを入力して変形するのに対し、図1(ロ)に示すネットワーク作成装置20は、文字列を入力し、それから新しいネットワークを作成する。

【0033】

【作用】前述した従来技術による手法において、図15に示すように余分なバスが生成されるのは、複数のネットワークを一つにマージする部分である。そこで本発明では、枝分かれのないネットワークを最後にマージして一つのネットワークにするという従来技術の手段を用いず、枝分かれもあり得る一つのネットワークに対して、直接、ルールを適用する。その際、ネットワークの中でルールによる変形の対象となる部分を、変形対象のネットワークから一旦分離し、その部分の変形を行った後、再び元のネットワークと合成する。

【0034】書き換えルールによる変形対象部分だけを分離して変形し、その変形結果と残りのネットワークとをマージすることにより、余分なバスの生成が行われなくなり、また変形が不十分な状態で終わることはない。

【0035】

【実施例】本発明で扱うネットワークは、例えば図2の(イ)に示すようなデータであり、データの基本単位(例えばA、B、…)を接続したものである。図2の(イ)に示すネットワークは、装置内では、同図(ロ)に示すようなデータ構造で管理される。各ノードの位置を示すノード番号に対応して、ノード名(または実データ)、前のノードのノード番号、後ろのノード番号に関する情報を持つ。

【0036】図1に示すルール記憶部13またはルール記憶部24に記憶する書き換えルールは、例えば図3の(イ)に示すようなルールである。この例のルールは、書き換え元が「A、B、/E、\_F、」であり、書き換え先が「C、」および「D、」である。これは、「E、」と「F、」に挟まれた「A、B、」があったならば、それを「C、」と「D、」に書き換えることを意味する。

【0037】このようなルールを、装置内では、図3の(ロ)に示すような構造で管理する。すなわち、変形対象パターンとして「A、B、」を、前後のコンテキストとして、それぞれ「E、」、「F、」を、変形後パターンとして「C、」および「D、」を記憶する。一般には、もっと単純な形のルールが多いが、そのデータ構造については同様である。

【0038】次に本発明による変形例を、図面を参照しながら説明する。図4の(イ)に示すnet9を、rule6に従って変形すると、「D、」が「Y、」と「Z、」に変換されることから、net21になる。これをさらにrule10により変形する。

【0039】まず、図1に示す入力部11に、net21が入力され、部分バス分離部12に渡される。ここでは、ネットワーク中でルールにマッチする部分バスを分離する。net21において、rule10の書き換え元にC、Y、がマッチするので、その部分だけを分離すると、net22のようになる。

【0040】次に、これが図1に示す部分バス変形部14に渡り、ルールに従って変形される。rule10によると、C、Y、にマッチした部分のC、を、U、とX、に変形することになっている。ルールにマッチした部分バスはすでに分離しており、その部分は枝分かれがないバスになっているので、あとは単純に部分バスを置き換えればよい。その結果は、net23のようになる。

【0041】図4の(イ)におけるrule10が、例えば同図の(ロ)に示すrule11のようであったとする。net21にrule11を適用すると、net24のようになる。ここで、図1に示すネットワークマージ部15にnet24が渡され、ネットワーク中の共通部分のマージが行われる。その結果、共通部分であるC、がマージされ、net25ができる。ネットワークマージ部15は、このようなマージを行う。

【0042】図1に示す部分バス分離部12は、例えば順方向部分バス分離部16と逆方向部分バス分離部17とを備えている。例えば、図5の(a)に示すnet13から、部分バスX、C、Z、を分離する場合について説明する。

【0043】部分バスの分離を前方からの分離と後方からの分離とに分けて行うことにより、分離を適切に行うことができるようにしている。ここでは、特に順番は規定しないが、まず前方からの分離を行ったとすると、結果はnet26のようになる。前方からの分離というのは、バス上のノード(X、C、Z、)を前から順に見ていき、左方向に出ている枝を部分バスの一部であるものと、そうでないものとに分けて、それぞれを別のノードで表すことである。

【0044】図6にその分離の過程を示す。例えば図6の(a)に示すようなnet28から、X、B、C、のバスを前方から分離する場合、(b)から(c)に示すような過程を経る。すなわち、(b)のnet29から(c)のnet30のように、分離したい枝(左方向への枝)以外に右側への枝がある場合には、二つに分けたノードの双方に、右側への枝をコピーする。

【0045】このような分離を、図5の(a)に示すnet13に対して行うことにより、図5の(b)に示すようなnet28になり、さらにnet26に対して、後方からの分離を行うと、net27のような結果となる。以上により、部分バスX、C、Z、の分離が完了する。

【0046】図1の(イ)に示すネットワークマージ部15(図1(ロ)のネットワークマージ部26も同様)は、ネットワークのマージを前方ノードから順に行う順方向ネットワークマージ部18と、後方ノードから順にマージする逆方向ネットワークマージ部19を直列に配置することにより、特に部分バスについてのネットワークのマージを行う。なお、順方向ネットワークマージ部18と逆方向ネットワークマージ部19を配置する順序は、どちらが先でもよい。

【0047】ネットワークマージ部15の説明のために、部分バス分離部12の説明で用いた図6に示すネットワークを逆にマージする過程について、図7に従って説明する。まず、図7の(a)に示すnet30のX、B、C、のバスを後ろからマージする場合について考える。このバスの直後には、D、とY、があるので、それぞれの前方の枝について、C、を探す。そうすると、二つ見つかるので、それらが一つのノードにマージできるかどうかをチェックする。チェックの結果、マージできなければ、次にB、についてチェックし、最後にX、に到達すれば終了である。

【0048】さて、ここで二つのC、がマージ可能かどうかを見るのであるが、二つのノードがマージ可能である条件は、以下の2条件である。

(1) 二つのノードの種類が同じ(ここでは、両方とも

C. である)。

(2) 後方(順方向マージの場合には、前方)に出ている枝の接続先がすべて同じ(ここでは、両方とも同じD.とY.である)。

【0049】以上の条件を満たしたので、C. はマージできる。マージするためには、一つのノードに両方の枝を接続し、同じ枝は一つにまとめる。その結果、図7の(b)に示すnet31ができる。同様に、B. もマージ可能である。X. は共通のノードがないためマージできない。したがって、逆方向ネットワークマージ部19の出力は、図7の(c)に示すnet32のようになる。

【0050】この結果は、さらに順方向ネットワークマージ部18へ渡されることになるが、net32はこれ以上マージできないため、このままネットワークマージ部15の出力となる。

【0051】図1の(ロ)に示すネットワーク作成装置20は、文字列入力部21により文字列を入力し、それを単純なネットワークに変換した後、ネットワーク変換装置10と同様な機能により、目的とするネットワークを自動作成する。図1の(イ)に示すネットワーク変換装置10における入力部11の代わりに、文字列入力部21とネットワーク変換部22が入った構成である。

【0052】入力は、単語文字列のような文字列であり、例えば図8の(a)に示すようなものである。これをネットワークの形式に書き換えると、同図(b)のようになる。このネットワークが、書き換えルールに従って変形され、最終的に変形後のネットワークとして出力される。

【0053】図1の(ロ)に示す文字列入力部21は、計算機の記憶装置(例えばディスク装置)に、単語文字列が記憶されているもので実現できる。ネットワーク変換部22は、文字列をネットワークに書き換えるものであり、ここでは、例えば前述した図2の(ロ)に示すようなデータ構造のネットワークに書き換える。すなわち、各ノードごとにノード番号とノード名が記述され、さらに前後に接続されるノードの番号が記述される。前後のノード番号が自分自身を指している場合は、ネットワークの終端であることを示す。ノード名は、長さが2の文字列で表されている。

【0054】単語文字列をネットワーク構造に変換する際には、ネットワークは枝分かれのないものになるので、次のようなアルゴリズムでネットワーク変換部22の機能を実現することができる。

【0055】(1)  $cn=1$ ,  $pn=1$ とする。

(2) 文字列から2文字(ノード名)を読み込む。文字列の終端であれば、処理(8)へ進む。

(3) 『ノード名』フィールドの $cn$ 番目へ、ノード名を書き込む。

(4) 『ノード番号』の $cn$ 番目へ $cn$ を書き込む。

(5) 『前のノード』の $cn$ 番目へ $pn$ を書き込む。

(6) 『後ろのノード』の $cn$ 番目へ $cn+1$ を書き込む。

(7)  $pn=cn$ ,  $cn=cn+1$ とし、処理(2)へ戻って処理を繰り返す。

(8) 文字列の終端が検出されたならば、『後ろのノード』の $pn$ 番目へ $pn$ を書き込む。【終了】

ここで、二重括弧で囲まれたものは、図2の(ロ)に示すデータ中のフィールド名を表している。

【0056】図1の(イ)に示す部分バス分離部12または図1の(ロ)に示す部分バス分離部23は、図9に示すように構成される。マッチング部90で、ネットワーク中で書き換えルールを適用できる部分を抽出し、その部分について順方向・逆方向の順で部分バスの分離を行う。

【0057】マッチング部90は、例えば図10に示す構成により実現される。まず、入力されたネットワークは、変形対象ボタンマッチング部91によって、変形対象ボタンが書き換えルールとマッチするかどうかを調べられる。なお、ここで用いる書き換えルールは、前述した図3の(ロ)に示すような構造をしている。マッチすれば、それが次段に渡される。マッチしなかった場合、そこでマッチングは失敗となり、全段をスキップして出力される。

【0058】同様に後方コンテキストマッチング部92および前方コンテキストマッチング部93により、書き換えるルール中の後方コンテキスト、前方コンテキストのマッチングが行われる。すべてマッチングすれば、そのネットワークおよびマッチしたノードの番号が出力される。どこかで失敗した場合には、そこで処理をストップして、マッチングに失敗した旨が出力される。

【0059】図10に示すルール分離部94は、書き換えルールを、変形対象ボタン・後方コンテキスト・前方コンテキストに分けて、各段に渡す処理を行うものである。書き換えルールが、例えば「(C.、D.)←A. B./E.、F.」である場合、変形対象ボタンマッチング部91に対して「A. B.」を、後方コンテキストマッチング部92に対して「F.」を、前方コンテキストマッチング部93に対して「E.」をそれぞれ分離して渡す。

【0060】マッチング部90でルールとマッチした場合には、その部分バスを、順方向部分バス分離部16および逆方向部分バス分離部17によって、前述したように分離する。この後、図1に示す部分バス変形部14(25)によって、部分バス分離部12(23)が分離した部分バスを、書き換えルールに従って変形する。具体的には、分離された部分バスのうち、変形対象ボタンにマッチングした部分を、書き換えルール中の『変形後ボタン』に記述されているボタンに置き換える処理を行う。

【0061】変形後ボタンに置き換えた部分バスは、図



1に示すネットワークマージ部15(26)による図7で説明した処理によって元の残りのバスとマージされる。

【0062】本発明を例えば音声認識装置等で使用する音響セグメントネットワークの作成に適用することにより、効率のよいネットワークの作成が可能になる。しかし、これに限られるわけではなく、データの基本単位を接続したネットワーク構造を持つデータであれば、同様に本発明を応用することが可能であることは言うまでもない。

【0063】

【発明の効果】以上説明したように、本発明によれば、冗長性の少ない適切なネットワークを効率よく作成することができ、そのネットワークを用いる装置の記憶容量の削減および処理速度の向上が可能になる。

【図面の簡単な説明】

【図1】本発明の原理説明図である。

【図2】本発明の実施例に係るネットワークのデータ構造を示す図である。

【図3】本発明の実施例に係る書き換えルールのデータ構造を示す図である。

【図4】本発明の実施例によるネットワークの変形例を示す図である。

【図5】本発明の実施例に係る順方向、逆方向部分バス分離部の作用説明図である。

【図6】本発明の実施例に係る部分バス分離部の説明図である。

【図7】本発明の実施例に係るネットワークマージ部の説明図である。

【図8】本発明の実施例に係るネットワーク変換部の説明図である。

【図9】本発明の実施例に係る部分バス分離部の構成図である。

【図10】図9に示すマッチング部の構成図である。

【図11】音響セグメントネットワークを用いた音声認識システムの説明図である。

【図12】音響セグメントネットワークの例を示す図である。

【図13】従来技術の説明図である。

【図14】従来技術によるネットワークの合成説明図である。

【図15】本発明の課題説明図である。

【図16】本発明の課題説明図である。

【符号の説明】

10 ネットワーク変形装置

11 入力部

12 部分バス分離部

13 ルール記憶部

14 部分バス変形部

15 ネットワークマージ部

16 順方向部分バス分離部

17 逆方向部分バス分離部

18 順方向ネットワークマージ部

19 逆方向ネットワークマージ部

20 ネットワーク作成装置

21 文字列入力部

22 ネットワーク変換部

23 部分バス分離部

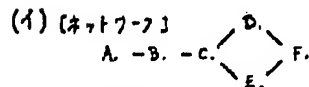
24 ルール記憶部

25 部分バス変形部

26 ネットワークマージ部

【図2】

ネットワークのデータ構造

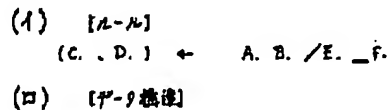


(ロ) [データ構造]

ノード番号	1	2	3	4	5	6
ノード名	A.	B.	C.	D.	E.	F.
前のノード	1	1	2	3	3	4
後のノード	2	3	4	5	6	6

【図3】

書き換えルールのデータ構造



変形対象パターン	A. B.
コンテキスト (前)	E.
コンテキスト (後)	F.
変形後パターン	C. D.

【図8】

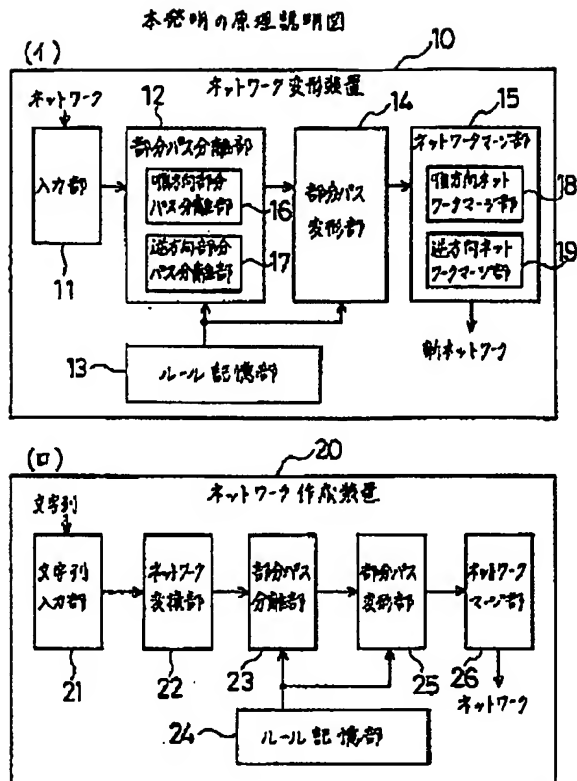
ネットワーク変換部の説明図

A. K. I. T. A.  
(Q) 入力文字列の例

A. -K. -I. -T. -A.  
(b) ネットワークの例

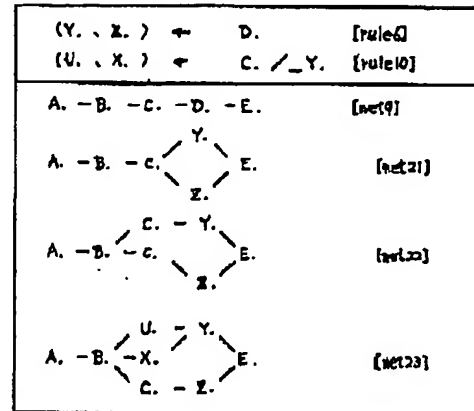


【図1】

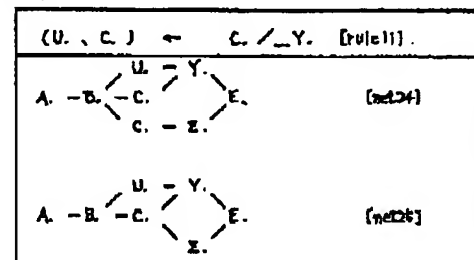


【図4】

(イ) 本発明による変形例

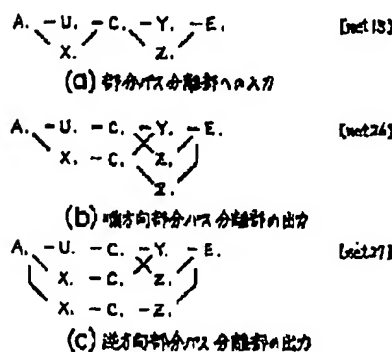


(ロ)



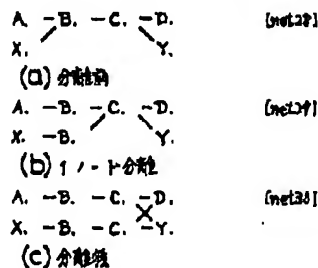
【図5】

順方向、逆方向部分パス分離部の作用説明図



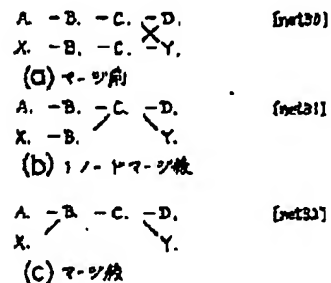
【図6】

部分パス分離部の説明図

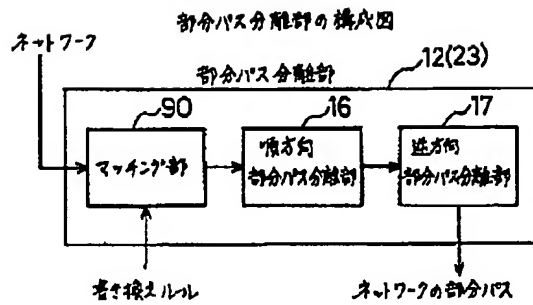


【図7】

ネットワークマージ部の説明図

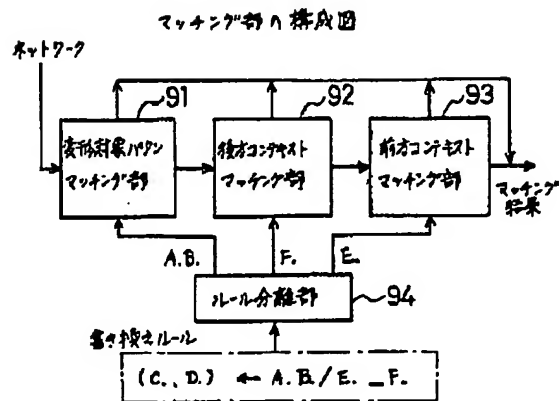


【图9】

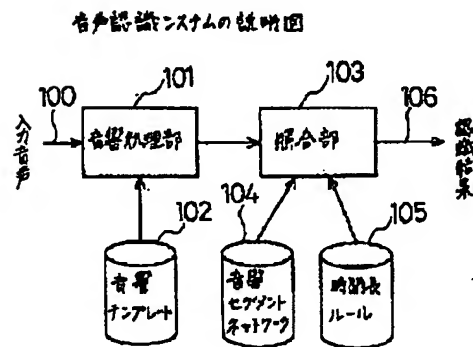


【图 1 1】

【~~☒~~ 1 0】

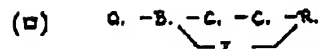
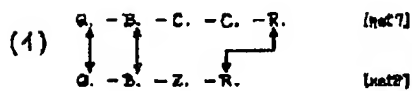


【图 12】

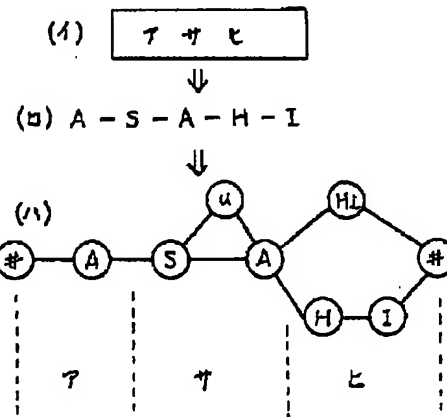


【图 14】

ネット7-7の合成説明図



### 合端セグメントネットワークの例



【例 15】

本発明の課題説明図

(Y, Z, )	←	D.	[rule6]
U.	←	B. / - C. Y.	[rule7]
X.	←	B. / - C. Z.	[rule8]

---

A. - B. - C. - D. - E.	[net9]
<pre>       U. - C. - Y.      /      \   A.          E.      \      /       X. - C. - Z.           </pre>	[net10]
A. - U. - C. - Y. - E.	[net11]
A. - X. - C. - Z. - E.	[net12]
<pre>       U.          Y.      /      \  /      \   A.          C.          E.      \      /  \      \       X.          Z.           </pre>	[net13]

【図13】

従来技術の説明図

(イ)

B.	← A.	[rule1]
B.	← A. / C. - R.	[rule2]
(B. , C. )	← A.	[rule3]
(B. , C. )	← A. / C. - R.	[rule4]
<hr/>		
Q. - B. - C. - A. - R.		[net1]
Q. - B. - C. - B. - R.		[net2]
Q. - B. - C. $\begin{matrix} \nearrow B. \\ \searrow C. \end{matrix}$ R.		[net3]

(ロ)

Z.	← C. B.	[rule5]
<hr/>		
Q. - B. $\begin{matrix} \nearrow Z. \\ \searrow C. \end{matrix}$ R.		[net4]
Q. - B. $\begin{matrix} \nearrow Z. \\ \searrow X - C. \end{matrix}$ R.		[net5]

(ハ)

Q. - B. - C. - B. - R.	[net6]
Q. - B. - C. - C. - R.	[net7]
Q. - B. - X. - R.	[net8]

【図16】

本発明の課題説明図

(イ)

A. - U. - C. - Y. - E.	[net11]
A. - X. - C. - Z. - E.	[net12]

(ロ)

A. - U. - C. - Y. - E.	[net11]
A. - X. - C. - Z. - E.	[net12]

(ハ)

(U. , X. )	← B.	[rule9]
<hr/>		
A. - U. - C. - Y. - E.		[net14]
A. - U. - C. - Z. - E.		[net15]
A. - X. - C. - Z. - E.		[net16]
A. - X. - C. - Y. - E.		[net17]

(ニ) [net14 + net15] :

A. - U. - C. $\begin{matrix} \nearrow Y. \\ \searrow Z. \end{matrix}$ E.	[net18]
--	---------

(ホ) [net16 + net16] :

A. $\begin{matrix} \nearrow U. \\ \searrow X. \end{matrix}$ - C. $\begin{matrix} \nearrow Y. \\ \searrow Z. \end{matrix}$ E.	[net19]
--	---------

(ヘ) [net19 + net17] :

A. $\begin{matrix} \nearrow X. \\ \searrow U. \end{matrix}$ - C. $\begin{matrix} \nearrow Y. \\ \searrow Z. \end{matrix}$ E.	[net20]
--	---------